

IN THE CLAIMS

1. [Previously Presented] A boot file generation method comprising:
 - creating a boot template file comprising symbolic variable names that point to configuration registers within a programmable device;
 - receiving user module selections with delineation of preferred configurations and functions associated with components of said programmable device;
 - generating application files automatically based upon user selected configurations and functions; and
 - substituting said symbolic variable names with actual configuration register names.
2. [Previously Presented] A boot file generation method of Claim 1 further comprising automatic interrupt vector mapping which assigns the appropriate providing interrupt processing routine vector.
3. [Previously Presented] A boot file generation method of Claim 1 further comprising:
 - providing an interface for selecting applicable user modules;
 - facilitating programming of desired functionality into the programmable device; and
 - executing an assembler process.

4. [Previously Presented] A boot file generation method of Claim 3 wherein said user module is a preconfigured function that may be based on more than one lock that work as a peripheral on the programmable device.

5.[Currently Amended] A boot file generation method of Claim 3 further comprising:
viewing and modifying user module parameters including setting global parameters;
specifying interconnections between the selected user modules; and
delineating a pin-out for each functional block making a connection between the software configuration and the hardware of the programmable device.

6. [Previously Presented] A boot file generation method of Claim 1 further comprising emulating the programmable device using an in-circuit emulator for debugging.

7. [Previously Presented] A boot file generation method of Claim 6 wherein the emulator allows the programmable device to be tested in a hardware environment while device activity is viewed and debugged in a software environment.

8. [Previously Presented] A boot file generation method of Claim 3 further comprising:
updating existing assembly source and C compiler code for device configurations; and
generating application program interfaces (APIs) and interrupt service routines (ISRs).

9. [Previously Presented] A boot file generation method of Claim 3 wherein the assembler process comprises operating on an assembly-language source code to produce executable code.

10. [Previously Presented] A boot file generation method of Claim 9 further comprising compiling and building code into an executable file.

11. [Previously Presented] A boot file generation method of Claim 9 further comprising linking programmed functionalities including device configuration.

12.[Currently Amended] A circuit comprising:

- a bus for communicating data;
- a microprocessor for processing data, said microprocessor coupled to said bus;
- a functional component coupled to said bus, wherein said functional component includes a plurality of functional blocks programmable to provide a plurality of functions and configurations; and

- a memory for storing configuration information including information associated with a boot file, wherein said boot file is created utilizing a template comprising symbolic register names that are substituted by the actual register name, said memory coupled to said bus.

13.[Cancelled]

14.[Original] The circuit of Claim 12, wherein a component of said circuit is programmable according to a configuration information stored in said memory.

15. [Previously Presented] A boot file creation method comprising:
- assigning variable symbolic register names to a user module;
 - establishing an association between the variable symbolic register names and actual configuration register names; and
 - replacing the variable symbolic register names with actual configuration register names.
16. [Previously Presented] The boot file creation method of Claim 15 further comprising loading the boot file on a target device.
17. [Previously Presented] A boot file creation method of Claim 15 further comprising utilizing a boot template to establish the association between the variable symbolic registers names and actual configuration register names.
18. [Previously Presented] A boot file creation method of Claim 17 further utilizing the boot template to replace the variable symbolic register names with the actual configuration register names.
19. [Previously Presented] A boot file creation method of Claim 15 further comprising regenerating the boot file is when a device configuration changes.
20. [Previously Presented] A boot file creation method of Claim 15 wherein the boot file is created and loaded on a target device by a design tool.